02 AOUT 1983

TRIUMF'S DATA ACQUISITION SYSTEM

T. Miles and A. Satanove*
TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C., Canada V6T 2A3

CM-P00067222

Summary

A flexible data acquisition system consists of a data acquisition task which is coupled to and under control of, a data analysis task. The data acquisition task is designed to interface to CAMAC through one or more Kinetic Systems 3912 Type A Crate Controllers to a Digital Equipment PDP-11/34, and it configures itself at the beginning of each run according to a user-written CAMAC definition file. The data analysis task is a modified version of the Fermilab Multi Channel Histogram Program. It controls the flow of the data acquisition task, and also serves as a multi channel pulse height analyzer.

Introduction

A flexible event-driven acquisition system, based on the Fermilab Multi Channel Histogramming Program¹ (MULTI) has been selected to be the site standard at TRIUMF. TRIUMF is a continuous duty, medium energy (500 MeV), high current (100 μ A) proton accelerator, which is used to conduct nuclear physics research involving various elementary particles. The typical nuclear physics experiment at TRIUMF involves acquiring data at a rate of several hundred 'events' per second, which is stored on magnetic tape for subsequent off-line analysis. Multi-channel histogramming is performed on a representative sample of the events on-line during the experiment, involving various cuts, and the on-line histograms are used to verify that the data being taped is satisfactory. This work (acquiring data, taping, histogramming) is conceptually the same for most experiments at TRIUMF, and a standardized hardware and software configuration has been developed (see Table 1).

Data Acquisition Task

The data acquisition task fetches the experimental data directly from CAMAC on an event-by-event basis, buffering it, and then writing buffers directly to tape. The CAMAC is interfaced to the PDP-11 via a Kinetic Systems 3912 type A crate controller. This controller assigns a unique memory location in the PDP-11's I/O page to every crate, station, and subaddress in the CAMAC system. Most CAMAC operations require only a single PDP-11 machine instruction (read or write) to the corresponding memory location. When an 'event' occurs, a module in the CAMAC crate (typically an Ortec C212 coincidence buffer) raises a flag known as a LAM (Look at Me), which causes the PDP-11 to 'vector' to an interrupt service routine, whose entry point is unique to the interrupting LAM. This interrupt service routine then proceeds to assign an 'event type', based on the entry point, an event 'serial number' to uniquely identify the event, before processing the CAMAC commands (defined in a disk file by the user), that make up the 'event list', clearing the LAM, and then dismissing the interrupt. If a buffer has been filled, a new buffer is requested, taping of the old buffer is initiated, and the buffer is sent to the Analysis (MULTI) task. Even though the site standard tape drives are fast (1600 BPI, 75 IPS), the data rate is generally limited by the taping speed. Therefore it is desirable to reformat the buffers to a larger size before taping them. This reformatting is also done by the data acquisition task, and a facility exists to perform a limited amount of event rejection during the reformatting process.

The specific characteristics of each experiment are stored in a disk file, which can be easily modified by

Table I
Standard TRIUMF Experimental Station

CPU:	PDP-11/34C (...also supported, PDP-11/60) with Floating Point Processor and Cache
Memory:	124K Words or 248 Kbytes
Storage:	Two cartridges of either Digital RL01 (5 megabyte), or Digital RL02 (10 megabyte) drives
Magtape:	1600 BPI 75 I.P.S. PERTEC, (two per machine) with Western Peripheral TC130 interface to PDP-11
Console TTY:	Decwriter II or Decwriter III
User TTYs:	Digital VT640 graphic crt (Modified Digital VT100). The modification makes terminal look like the Tektroniks 4010 graphics storage terminal
Plotting:	Texprint mod to Decwriter III / Printronix / Versatec
CAMAC interface	is provided by one or more Kinetics 3912 type A crate controllers. They are single-board, interrupt-driven devices that map CAMAC directly into PDP-11 memory
Oper. System:	Digital Equipment RSX-11 M multi tasking monitor
Acquisition:	TRIUMF-written data acquisition (DA) task
On-Line Anal.	Enhanced version of FERMILAB's Multiple Histogram Analysis Program (MULTI)

the user. This file defines the hardware configuration, and is read every time that data acquisition is initiated. It specifies what (if any) hardware initialization is to be performed, it indicates what LAMS are expected, and it associates an 'event list' with each active LAM. The 'event list' describes what CAMAC operations are to be performed whenever the LAM gets set, and the 'event list' allows for reads, writes, and non-data CAMAC functions, and whether the event length is fixed, or is of variable length (Q-Stop mode). The user is responsible for clearing the LAM, and writing any output pulse(s) necessary during the 'event list'. Typically, an output pulse is generated to a standard NIM (Nuclear Instrumentation Module) output driver (Ortec NDO27), by writing a logical .TRUE. to the driver at the start of the 'event list', and then writing a logical .FALSE. to the driver at the end of the 'event list'. The electronics of the experiment can then be configured so that no events are lost, by disabling events as soon as an event is detected in hardware, and keeping events disabled, until the trailing edge of the NIM output pulse appears (indicating that the computer is ready to process further events). A NIM pulse is also generated if there is no room to store the data from the event (because the data acquisition task is waiting for the magnetic tape) Event recognition is then deferred until a buffer becomes available.

All data transfers to and from CAMAC involve the PDP-11 central processing unit, because the Kinetics 3912 does not have the capacity to perform direct memory access

(DMA) to the PDP-11 memory. Also, all buffer management must be performed by the PDP-11 CPU, because the Kinetics 3912 does not possess any intelligence. The CPU overhead incurred has been estimated, and an empirical formula derived for calculating the CPU overhead per event. This formula (for a PDP-11/34 with cache) gives the CPU overhead N(uS.) as

$$N(uS.) = (10. uS/Line \text{ in Interrupt List}) * (\# \text{ Lines}) \\ + (3. uS/Dataless \text{ Function}) * (\# \text{ Dataless Functions}) \\ + (6. uS/16. \text{ bit Read}) * (\# \text{ 16. bit Reads}) \\ + (11. uS/24. \text{ bit Read}) * (\# \text{ 24. bit Reads}) \\ + (220. uS. \text{ overhead setting up/dismissing interrupt})$$

Additional overhead is incurred when the buffer is reformatted to a bigger buffersize for taping (to reduce tape wastage from inter-record gap), when the buffer is copied to the Analysis Task (MULTI), and by various housekeeping routines that the data acquisition task performs. By histogramming the program counter, the following estimate of the CPU requirements of the data acquisition task (normalized to 100%) was made

Interrupt Service Routine	64%
Reformatting Tape Buffer	30%
Copying Buffer for Analysis	4%
Miscellaneous	1%

This suggests that the value of N arrived from the above formula should be divided by 0.64, to estimate the total CPU time demanded by the data acquisition task.

A 'typical' TRIUMF nuclear physics experiment will now be considered, to try to estimate the CPU time required by the data acquisition task. In this 'typical' experiment, data from 4 CAMAC ADC converters was read into a buffer of size 1001 words, and then reformatted into a buffer of 4095 words for taping. Each ADC converter produced 13 words, there were five such converters to be read, and there was a fixed overhead on each event of five words (containing event length, event type, run number, serial number) to produce a total of 57 words per event. Using the above formula, the interrupt service time was calculated to be 0.59 mS, and actually measured to be 0.56 mS. Dividing the (0.56 mS) measured interrupt service time by the relative (0.64) factor to calculate the total overhead, an estimate (0.87 mS) was obtained for this overhead. With the 75 ips tape drives recording 4095 word blocks at 1600 BPI, the recording speed (53 words/mS) was calculated, including the effect of the (0.6 inch) interrecord gap. The maximum event rate recordable (0.93 events/mS) was then calculated, and the CPU overhead (0.81 mS/mS) estimated. There was a small, but finite percentage (19%) of CPU time remaining for the Analysis Task (MULTI).

Analysis Task

Fermilab's multi channel histogramming program (MULTI)¹ is used to interface the experimenter with the data acquisition. MULTI is a very flexible system, widely used in the nuclear physics community, and thus its selection tends to minimize the amount of 'folklore' that new experimenters have to learn. MULTI enables the physicist to Begin, End, Suspend, and Resume data acquisition, it enables the physicist to generate cuts, histograms, and scatterplots 'on the fly', and it permits both on-line and off-line analysis of experimental data.

The Fermilab multi channel histogramming system as used at TRIUMF actually consists of several cooperating tasks. The major task (MULTI) is always active, and is responsible for maintaining the experimental data base. The data acquisition task (DA) is active only when a run is in progress (see Fig. 1). Each user communicates to MULTI through a private copy of the command

Fig. 1. Command and data flow in real time.

queuer task (MUL). Whenever the user wishes to issue a MULTI command, he types the appropriate string into the command queuer task (MUL). When the command string has been completed, the command queuer requests a buffer from MULTI's common area, copies the command into the MULTI buffer, and causes MULTI to queue the command in its input stream. Before analysing the next event, MULTI checks to see if any new commands have been queued. If MULTI detects a command waiting for processing, the command is dequeued, processed and any type-out is directed to the appropriate terminal. This procedure enables several users at different terminals to simultaneously interrogate MULTI's data base.

The performance of the TRIUMF version of MULTI has been considerably enhanced over the standard (Fermilab Level 1.0) version. The histogram memory (dynamic area) has been extended to 32 Kwords through the use of RSX-11M programmed logical address space (PLAS) directives. Support for two-dimensional histograms has been added, now that there is sufficient histogram memory to support reasonable two-dimensional resolution. The histogram display routine GHIST has been rewritten so that it rebins all histogram display segments involving 512 bins or more. A special assembly-language routine has been written which allows MULTI to play back tapes with block sizes up to 4095 words. The (heavily used) variable access function GETVAL has been optimized for speed, and completely rewritten using the PDP-11 assembly language MACRO. Following the suggestion of K.C. Green² (formerly of the University of Washington, Seattle), the numerical analysis now compiles directly into PDP-11 machine code, the histogram definition blocks now reside in local (COMMON) storage, and visual updating of histograms is now conditional on the availability of the graphics device. These changes have considerably improved MULTI's rather slow analysis rate, as shown below:

	TRIUMF Version	Standard Version
Idle Loop	0.71 mS	1.1 mS
Eval Code IV4=IV4	0.07 mS	0.7 mS
Update Histogram	0.33 mS	1.1 mS
Visual Update Hist.	2.7 mS	25.0 mS

The dynamic memory management routine CORAL has been completely rewritten. Dynamic memory is now allocated from the beginning of the dynamic memory area, using a 'best fit' algorithm. When a free fragment of memory

is returned, CORAL now attempts to merge it with adjacent free fragments, before inserting it in the free fragment queue. CORAL now places the dictionary in a known offset from the beginning of dynamic memory, and this permits other (user-written) tasks to 'look-up' named objects which have been 'entered' in MULTI's dictionary.

User modifications to the MULTI program itself have been strongly discouraged at TRIUMF. Maintenance of a single site-standard MULTI version has been found advantageous for multiple reasons. Experimenters only require an executable image of the MULTI program, and hence only one version of MULTI needs to be maintained. If the software acts strangely, the cause is usually traceable to bugs within the MULTI code itself, and not the second-order effect of a user modification. Only one backup of MULTI needs to be maintained, because all the user copies are (in theory) identical, and up to date. MULTI itself is a very big, cumbersome program, with many overlays, and having very little virtual address space remaining. Therefore even seemingly simple changes to MULTI become non-trivial operations, frequently introducing obscure bugs, and generally requiring a lot of consultation and planning.

Functional enhancements to MULTI have been implemented as separate tasks under the RSX-11M operating system. These tasks, being independent of MULTI, have lots of virtual address space available, and do not usually require the use of an overlay structure. They usually are dedicated to a single function, are conceptually simple, and can be coded directly by the physicist himself in FORTRAN with a minimum of consultation. Programming errors are isolated to the faulty task by the memory protection features of the RSX-11M operating system, and thus do not affect the operation of MULTI. Access to MULTI's data base is performed through a set of FORTRAN-callable routines, which permit the user to read the contents of MULTI's dynamic memory.

Accessing MULTI's data base from an external task is a simple process. To locate a data base element by name, the user performs a 'lookup' in the MULTI dictionary (using FORTRAN-callable subroutines NAME). If the name is known to MULTI, the dictionary entry will contain a pointer to the data base element. The first word of the data base element will describe the element, indicating whether it is a display block, scatterplot, histogram, etc. If the element is a histogram, the next eleven words will constitute a histogram header, and they will describe the number of bins in the histogram, the bin width, the binned variable, the lower window, the number of underflows and overflows, and the histogram itself will directly follow the histogram header, stored as one PDP-11 word per bin.

Utilities

TRIUMF has also developed several general purpose utilities to ease the life of the experimenter. A tape utility (SPY) has been written which permits the experimenter to copy tapes, examine records in Ascii or Decimal, and write tape End of File marks (which may be missing from data tapes). Another utility (SCA) has been written, which analyses all scalars read during an experiment. (MULTI cannot do this because its analysis rate is too slow.) A histogram archive task (MULTI-SAVE) has been written to allow the experimenter to save selected one-dimensional and two-dimensional histograms on disk, in a standard format, for subsequent off-line analysis.

An external task GRAPH has been written to process histograms, either directly from MULTI's data base, or from disk images created by the histogram archive task MULTISAVE. GRAPH has facilities for displaying

histograms on Tektronics 4010 video terminals, or plotting histograms on Printronix or Versatec hard copy devices. GRAPH contains a selection of useful numerical routines which calculate gaussians, sum counts, calculate centroids, and other moments. GRAPH interacts with its own command querier task (GPH), in a manner similar to the way MULTI interacts with its command querier task (MUL).

Limitations of Current Configuration

In the present configuration of the TRIUMF data acquisition system, the data acquisition task has a higher priority, and can process events much more rapidly, than the analysis task MULTI. As shown above, the data acquisition task can easily consume the majority, if not all, of the CPU time. (To prevent the data acquisition task from locking out the CPU entirely, it has been modified to relinquish a minimum of 10% of the CPU time to other concurrent tasks.) The performance of the analysis task MULTI is thus degraded for on-line data acquisition, and typically processes events at only 100 Hz or so. Thus only a small fraction of the events taped by the data acquisition task get analysed on-line, and the MULTI event analysis serves mainly as a consistency check for the taped data. MULTI is used on-line mainly because our physicists are familiar with its operating environment, utilizing MULTI commands to indirectly control the operation of the data acquisition task. MULTI is also used off-line to 'play back' taped data, (see Fig. 2) and in this mode all events are analysed, and the analysis rate is much improved (no CPU competition with the DA task), but our physicists seem to prefer analysing data tapes off-line using the more sophisticated programs available on our Digital Equipment VAX-11/780.

Future Expansion

The functionality of the existing experimental systems is to be enhanced in the near future. The histogram memory, which is currently on the UNIBUS, will be moved to a special 'dataspace' memory device, and this will free up 32K of UNIBUS address space, as well as increase the speed of histogram access. The additional memory released will enable the support of RSX-11M version 4.0, with its DCL command interpreter. There will also be sufficient memory to support the full-duplex terminal driver, and this will allow the physicists to use the visual editor EDT. At some time, the systems may be linked into DECNET via DL-11 serial interfaces, and this will allow the physicists remote

Fig. 2. Command and data flow for off-line analysis.

access (via telephone) to their experiments, as well as giving them the capability of file transfer. One major concern with DECNET was the large overhead (30% - 50% of the CPU time) incurred when active at 9600 baud. Another concern with DECNET was the rapid interrupt response (less than 0.5 millisecond) required on the DL-11, which necessitated the rewriting of the data acquisition (DA) task so as to execute at lower (level 5) hardware priority during the lengthy interrupt service routines.

A possible acquisition system contemplated for TRIUMF in the future involves the use of an intelligent front-end processor, which would control the experimental hardware (CAMAC or FASTBUS), and would have direct memory access capability to a 32. bit mainframe computer. This configuration would ideally off-load all critical real-time processing (event rejection, buffering, queue management) onto the (fast) front-end processor, leaving the 32. bit mainframe free to devote 100% of its CPU time to event analysis and control

functions, with no real-time constraints. One configuration would see the experimenter programming the data acquisition code for the experiment in a high level language on the mainframe, which would then compile this code and down-line load it into the front-end processor at the start of the experiment. The analysis program (customized by the experimenter) would be written in FORTRAN, and would operate without addressing or real-time constraints, since it would execute on its own dedicated 32. bit mainframe.

References

1. J.F. Bartlett, J.R. Biel, D.B. Curtis, R.J. Dosen, T.D. Lagerlund, D.J. Ritchie and L.M. Toff, "RT/RSX MULTI: Packages for Data Acquisition and Analysis in High Energy Physics", IEEE Trans. Nucl. Sci. NS-26, No. 4, 4633-4635, August (1979).
2. K.C. Green, "Speed Optimizations for Fermilab MULTI", IEEE Trans. Nucl. Sci. NS-28, No. 5, 3858-3862, October (1981).

*Present address: Epic Data Systems, 7280 River Road
Richmond, B.C., Canada V6X 1S5